Computer Integrated Surgery II Paper Critique
Group 1: Motorized Fixation to Tubular Retraction in Brain Surgery
Robby Waxman
Rwaxman5@jhu.edu
3/04/21

Statement of my Project

My project is “Motorized Fixation to Tubular Retraction in Brain Surgery”. The objectives of this project are to create a mechanical device that will stabilize the tubular brain retractor during a surgical procedure, as well as to create software that will allow for the realignment of the tubular retractor from two main sources during the procedure. The first, simpler source would be moving the tubular retractor’s angle of orientation based on a joystick controlled by the surgeon. Once this has been accomplished, we aim to create a method of realignment based on the orientation of the surgeon’s forceps. This way, when the surgeon presses a button, the tubular retractor will be realigned to match the same orientation as the forceps at the time the button was pressed.

Paper Selection

The paper I have selected for my critical review is *Algorithm for Estimating the Orientation of an Object in 3D Space, Through the Optimal Fusion of Gyroscope and Accelerometer Information* (Contreras-Rodriguez 2017). This paper contains the mathematics for three different methods of orientation estimation based on Inertial Measurement Unit (IMU) data. Additionally, it analyzes the performances of the three different methods in a series of static and dynamic experiments.

Summary of Goal, Key Results, and Significance

Inertial Measurement Units (IMUs) are cheap sensors that contain a gyroscope and an accelerometer and are often used to address the problem of orientation estimation. Gyroscopes measure the angular velocity of an object, and naturally from this, if we calibrate a gyroscope then the orientation of an object should be able to be estimated. However, in practice, orientation estimates provided by gyroscopes alone are subject to large errors caused by drift and offset. The goal of this paper was to implement a method to correct for errors caused by both drift and offset by the fusion of gyroscopic data with accelerometer data from the same IMU to improve orientation estimation.

The authors implemented three different methods of combining the data to accurately estimate orientation. The methods had varying levels of computational complexity and performance. However, two of the three methods were able to rather successfully estimate the orientation in both static and dynamic scenarios far more accurately than relying solely on gyroscope data.

These results were significant because they presented a straightforward approach to estimating orientation that can be completely in a computationally efficient manner. Much of the prior literature either relied on gyroscope data alone, complex computational models, or additional sensors to provide accurate estimates of orientation. In settings like ours, we need close to instant feedback (efficient computation) and minimal sensors as they interfere with a surgeon's ability to properly utilize forceps and other tools.

Necessary Background

It is not necessary to have knowledge of any prior work performed by the authors, but it is important to understand some of the basic terminology used in this paper.

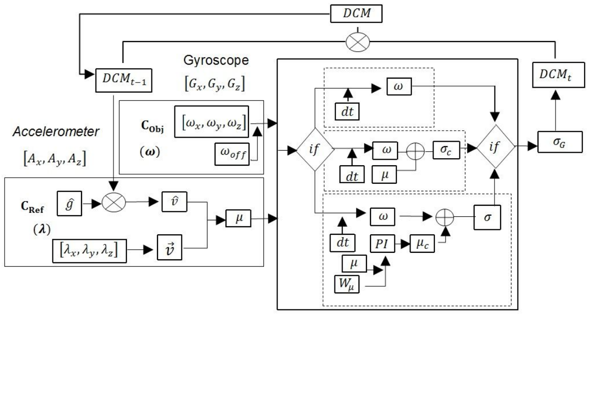
For example, it is useful to know the differences between offset and drifting and how they both affect the data readings of sensors. In simplest terms, offset is the deviation between the reference position and the zero position. On the other hand, as the paper mentions, drift is caused by "integration of gyroscope measurement errors" and this leads to an accumulating gyro drift.

Additionally, it is useful to be familiar with direction cosine matrices (DCM) which are essentially frame transformations between Cartesian bases.

Technical Approach

Overview

Below is a block diagram of the entire system. It includes the calculations for all 3 orientation estimation methods. The goal of this system is to reduce the error in gyroscope orientation estimations by incorporating data provided by the accelerometer that is already collecting data inside of the IMU.



(Contreras-Rodriguez 2017)

Proposed Algorithms

The first proposed algorithm estimated the orientation σ_G by including the offset correction ω_{off} in ω (the mathematics for the offset correction will be explained later). This algorithm can be seen represented in the block diagram above in the upper if statement in the right block. The second proposed algorithm estimated the orientation as σ_C by including both ω_{off} and μ - a correction vector for drift calculated from the accelerometer output. Lastly, the third algorithm estimated the orientation as σ using a PI feedback controller in order to update the gyroscope's estimated orientation to remove drift and reduce noise. All 3 of these methods and their procedures for computation can be seen in the block diagram above.

Offset Correction

Offset is corrected for in all 3 of the algorithms proposed in the paper before estimating orientation. This calculation is performed by computing the gyroscope offset and removing this from the gyroscope reading. This was done using the math below, in this case they take 1000 samples of data to calculate ω_{off} but it appears "n" can be arbitrarily large number of samples.

$$\omega_{off} = \frac{\sum_{i=0}^{n} \omega}{n}$$

$$\omega = \omega - \omega_{off}$$

(Contreras-Rodriguez 2017)

Where the resulting ω (one on the left hand side) is the corrected orientation. However, while the math makes sense, I would just like to add that they could have used better notation such as defining the corrected orientation ω a different symbol from the uncorrected orientation ω.

Drift Correction

Drifting is corrected for in the second and third algorithm proposed, but not the first. The way in which the drift estimation is integrated back with the gyroscope data varies between the second and third algorithm, however the way in which drift is calculated is the same between the methods. Drifting was calculated by relating the the inertial frame C_{ref} to a local level coordinate system λ (from the accelerometer data). Then, taking into account gravity "g", a rotational correction vector μ was estimated using the formula below:

$$\vec{v} = \frac{\lambda}{|\lambda|}$$
$$\hat{\mathrm{v}} = \hat{g} * DCM_{G_t}$$
$$\vec{\mu} = \vec{v} \times \hat{v}$$

(Contreras-Rodriguez 2017)

In the above math DCM stands for Direction Cosine Matrix which, as mentioned before, is essentially a matrix to convert vectors between different bases. In their methods, the authors actually choose to use an approximation of the DCM to increase computational efficiency. The derivation of their approximation is shown below:

$$\begin{bmatrix} C(\theta)C(\phi) & -S(\theta)C(\alpha) + C(\theta)S(\phi)S(\alpha) & S(\theta)S(\alpha) + C(\theta)S(\phi)C(\alpha) \\ S(\theta)C(\phi) & C(\theta)C(\alpha) + S(\theta)S(\phi)S(\alpha) & -C(\theta)S(\alpha) + S(\theta)S(\phi)C(\alpha) \\ -S(\phi) & C(\phi)S(\alpha) & C(\phi)C(\alpha) \end{bmatrix}$$

$$\begin{matrix} \cos(\sigma) \approx 1 \\ \sin(\sigma) * \sin(\sigma) \approx 0 \end{matrix} \quad Assuming \begin{cases} \sin(\alpha) \\ \sin(\phi) \\ \sin(\theta) \end{cases} \approx \begin{bmatrix} \alpha \\ \phi \\ \theta \end{bmatrix}$$

$$DCM = \begin{bmatrix} 1 & -S(\theta) & S(\phi) \\ S(\theta) & 1 & -S(\alpha) \\ -S(\phi) & S(\alpha) & 1 \end{bmatrix} = \begin{bmatrix} 1 & -\theta & \phi \\ \theta & 1 & -\alpha \\ -\phi & \alpha & 1 \end{bmatrix}$$

$$DCM = \begin{bmatrix} 1 & -\theta & \phi \\ \theta & 1 & -\alpha \\ -\phi & \alpha & 1 \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$[\epsilon_x, \epsilon_y, \epsilon_y] = [r_{xx}, r_{xy}, r_{xz}] \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} = X * Y^T$$

$$[\mathrm{X}', \mathrm{Y}', \mathrm{Z}'] = \left[X - \frac{[\epsilon_x, \epsilon_y, \epsilon_z]}{2} Y, Y - \frac{[\epsilon_x, \epsilon_y, \epsilon_z]}{2} X, (\mathrm{X}') \times (\mathrm{Y}')\right]$$

$$DCM = \begin{bmatrix} X_{norm} \\ Y_{norm} \\ Z_{norm} \end{bmatrix} = \left[\frac{\mathrm{X}'}{|\mathrm{X}'|}, \frac{\mathrm{Y}'}{|\mathrm{Y}'|}, \frac{\mathrm{Z}'}{|\mathrm{Z}'|}\right]^T = \begin{bmatrix} 1 & -\theta & \phi \\ \theta & 1 & -\alpha \\ -\phi & \alpha & 1 \end{bmatrix}$$

(Contreras-Rodriguez 2017)

Data Integration

In the second proposed algorithm, the integration of the drift compensation is quite simple and straightforward. The correction vector is simply fed back into the gyroscope orientation (with the offset corrected) as such:

$$\sigma_c = \sigma_G + \vec{\mu}$$

(Contreras-Rodriguez 2017)

In the third proposed algorithm however, the integration of the drift compensation is done through a Proportional Integral Feedback Controller (PI). With the PI, the rotational correction vector is multiplied by a weight and fed to the PI controller. The purpose of this is to supposedly converge the two coordinate systems (reference and object). Then the adjusted correctional vector is added back in. The math is shown below:

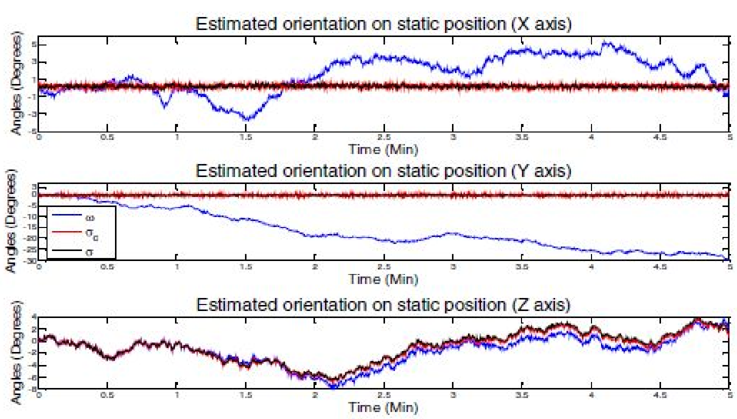
$$\mu_c = K_p f(W_\mu * \mu) + K_i \int_0^t f(W_\mu * \mu)\, dt$$
$$\sigma = \sigma_G + \mu_c$$

(Contreras-Rodriguez 2017)

Experimental Evaluation and Results

Static Experiment

In the static experiment, the IMU is held steady for a period of 5 minutes (300 seconds) and the three different proposed algorithms are all separately used to measure the orientation throughout the full 5 minute period. Since the IMU was held steady and flat, the estimated orientation should be an angle of 0 in the X, Y, and Z planes. The estimated angles in each plane over the 5 minute period are shown below for each method:

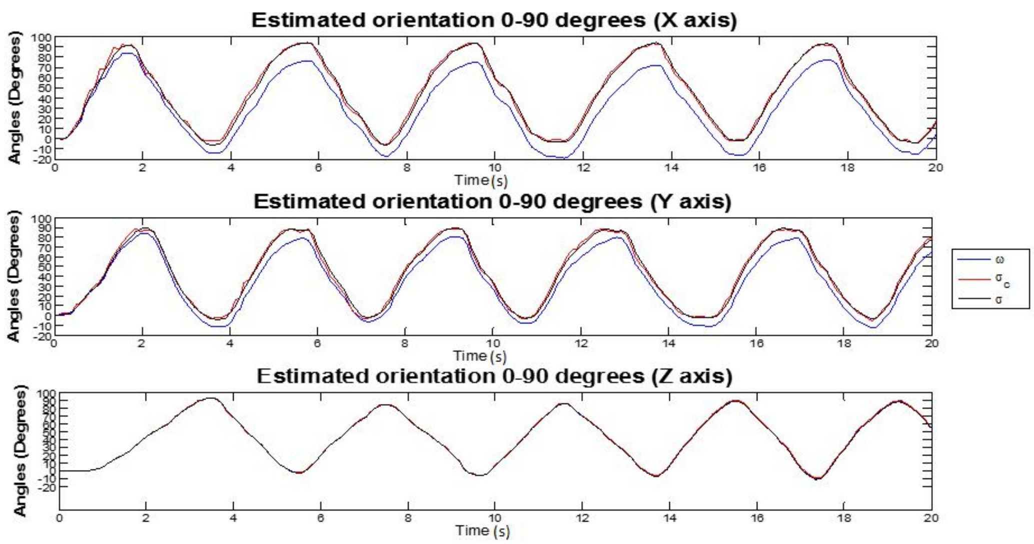


(Contreras-Rodriguez 2017)

For this diagram, the blue line represents the first method, the red line represents the second method, and the black line represents the third method. As we see, the second and third methods tend to do a much better job of orientation estimation in the X and Y planes, but all three methods struggle to properly capture the Z plane. This is consistent with previous literature as in the Z plane, the IMU must try to balance the effects of gravity. As we see here, the second and third methods appear to properly reduce the effects of drift and increase accuracy.

<u>Dynamic Experiment</u>

In the dynamic experiment, the IMU is rotated in a set of five repetitions from 0 to 90 degrees over the course of 20 seconds and the three different algorithms are all separately used to measure the orientation throughout the full 20 second period. For this procedure, we have an intuitive idea of what the correct curve should look like, but that information is never provided in the plot or in the paper and is a point of criticism from me. However, the three different method's estimates are plotted together for comparison. The results are shown below:



(Contreras-Rodriguez 2017)

For this experiment, we see marked improvement in terms of estimating orientation with the second and third methods in the X and Y axis, but no significant difference in the Z axis which is consistent with the static experiments.

<u>Assessment</u>

This paper was important as it provided a computationally efficient method of improving orientation estimation by incorporating data from the IMU's accelerometer. Therefore, it has potential in improving orientation estimation in low resource, real time settings, or limited spaces where we may not have the money, time, or space to include additional sensors to supplement our IMU or run more complicated architecture. It is relevant to my project because it provides a good basis for the development of our own orientation estimation algorithm that will be used for the realignment step, as well as provides insight on what we may struggle with as we develop our algorithm (mainly estimation in the Z axis).

There are several good and bad points made in this paper. First, the paper provides mathematical explanations for their approaches and the justifications are sensible for the most part. However, there are times where variable naming conventions are unclear. Furthermore, little work is done to justify why a PI controller is the right method to correct for high noise during drift correction. That being said, the methods do appear to have a marked improvement over the standard gyroscope data or even gyroscope with offset. This leads me to my next criticism, it is unclear how well their algorithm performs quantitatively in both the static and dynamic experiment as they never provide numerical measurements throughout their analysis. It would have been helpful to maybe provide mean degrees error or something along those lines. Lastly, my final criticism revolves around the dynamic experiment. They simply plot the three curves of the estimates without also plotting the known orientation at that time. I think this plot would have made much more sense if they instead plotted the quantity of the Estimate - Real Value and then it would be much clearer how far the estimates were off.

A suggestion for future work is extending the algorithm to incorporate data readings from multiple IMUs. Additionally, another future experiment should be repeating the dynamic experiment for different movements and longer durations as well.

Conclusion

This paper introduced several efficient ways of increasing the accuracy for orientation estimation using only an IMU. It provides a lot of insightful mathematics for performing orientation estimation, especially in our application in which we need high efficiency and cannot place additional sensors onto the forceps. However, the experiment section was lacking without additional trials, quantitative error measurements, and justification. We will most likely use this as a basis for our algorithm, and make improvements where we feel fit, hopefully fixing the issues with the Z-axis that the algorithm could not capture.

Reference

Contreras-Rodríguez L.A., Muñoz-Guerrero R., Barraza-Madrigal J.A, “Algorithm for Estimating the Orientation of an Object in 3D Space, Through the Optimal Fusion of Gyroscope and Accelerometer Information.” 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico. September 20-22, 2017